

REDES NEURAIS COM TENSORFLOW



CLASSIFICAÇÃO: NEURÔNIO SIGMÓIDE

DIEGO RODRIGUES DSC

INFNET

CRONOGRAMA

Dia	Aula	Trab
29/07	Perceptron de Rosenblatt	
31/07	Classificação: Neurônio Sigmóide	
05/08	Classificação: Rede Neural Feedforward	Grupos
07/08	Classificação: Treinamento Robusto	
12/08	Regressão	Base de Dados
14/08	Agrupamento	
19/08	Séries Temporais	Modelos
21/08	Apresentação dos Trabalhos Parte I	

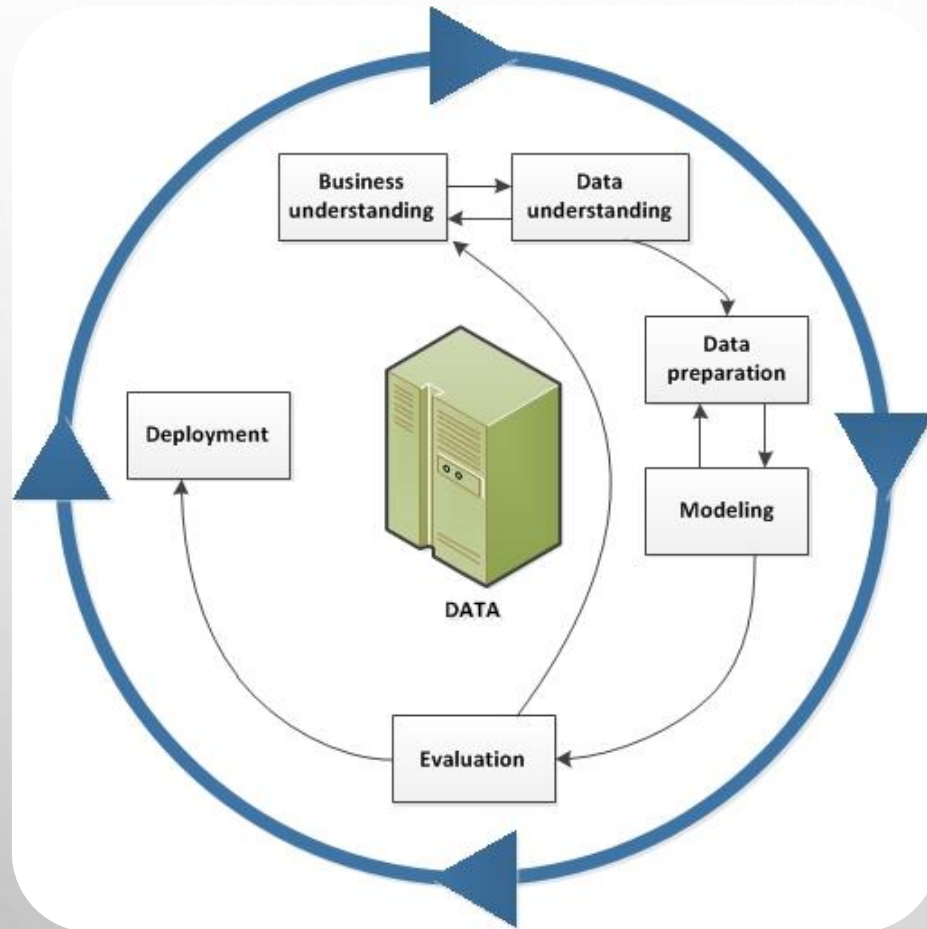


CLASSIFICAÇÃO : NEURÔNIO SIGMÓIDE

- PARTE 1 : TEORIA
 - METODOLOGIA CRISP
 - APLICAÇÃO AOS DADOS IRIS
- PARTE 2 : PRÁTICA
 - NOTEBOOK: CLASSIFICADOR IRIS
- PARTE 3 : TRABALHOS
 - ESCOPO & EVOLUÇÃO

The background of the slide is a light gray gradient. In the top-left and bottom-right corners, there are clusters of realistic water droplets of various sizes, rendered with soft shadows and highlights to give them a three-dimensional appearance. In the center of the slide, there is a faint, circular watermark. The watermark contains a stylized graphic of a person with arms raised, surrounded by text that is not clearly legible but appears to be in Italian.

PARTE 1 : TEORIA

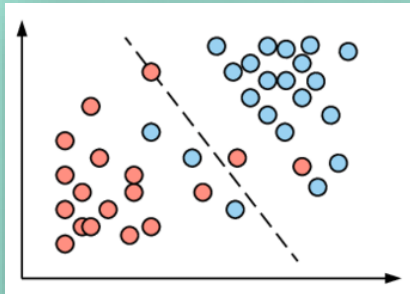


CROSS INDUSTRY PROCESS FOR DATA MINING (CRISP-DM)

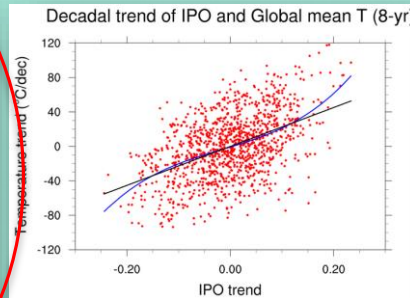
APRENDIZADO SUPERVISIONADO

Tarefas de **classificação** e **regressão** pertencem a esta categoria. O treinamento consiste em **encontrar parâmetros** para o modelo que **minimiza uma função de risco/erro** para uma amostra de treinamento, baseado na diferença entre os **valores previstos e reais**, para cada observação.

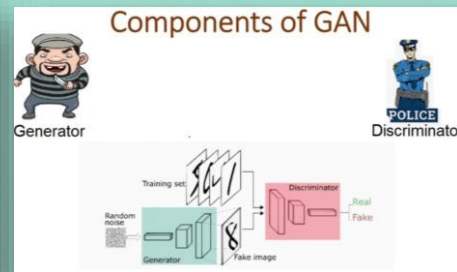
APRENDIZADO SUPERVISIONADO



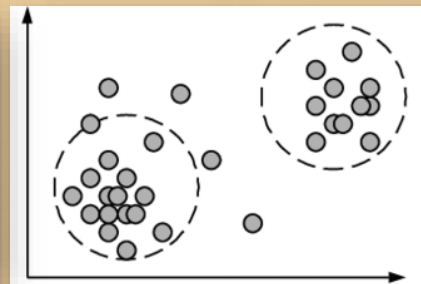
CLASSIFICAÇÃO



REGRESSÃO



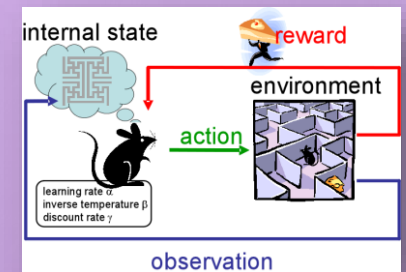
GENERATIVO



AGRUPAMENTO

APRENDIZADO NÃO- SUPERVISIONADO

APRENDIZADO POR REFORÇO



REFORÇO

CLASSIFICAÇÃO

Um bebê consegue separar e ordenar blocos com diferentes tamanhos, formas e cores. Ele também consegue **identificar os tipos diferentes de objetos**.

Os diferentes tipos de objetos são chamados de **classes**. As características dos objetos são chamadas de **variáveis** ou **atributos**.



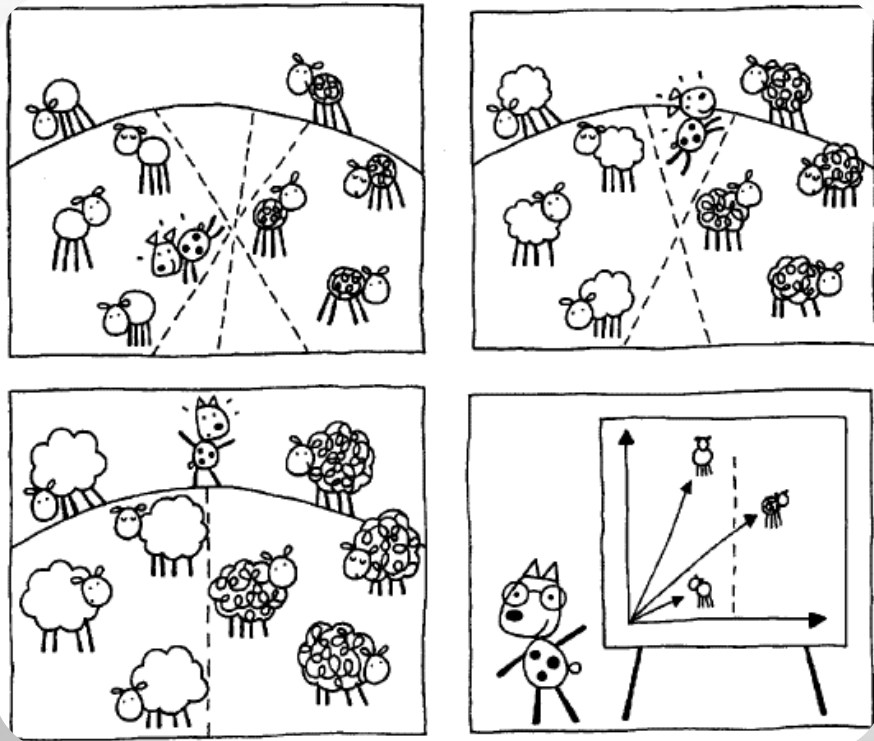
Data Scientist da Nova Geração

Então, um classificador é um modelo **treinado para discriminar objetos** pertencentes a duas ou mais classes, baseado em seus atributos.

The slide features a light gray gradient background. In the top-left and bottom-right corners, there are clusters of realistic water droplets of various sizes, rendered with soft shadows and highlights. Faintly visible in the upper center is a circular logo, which appears to be the United Nations emblem.

DATA UNDERSTANDING & PREPARATION

REPRESENTAÇÃO



Ideia: como quantificar um objeto no mundo físico no mundo digital?

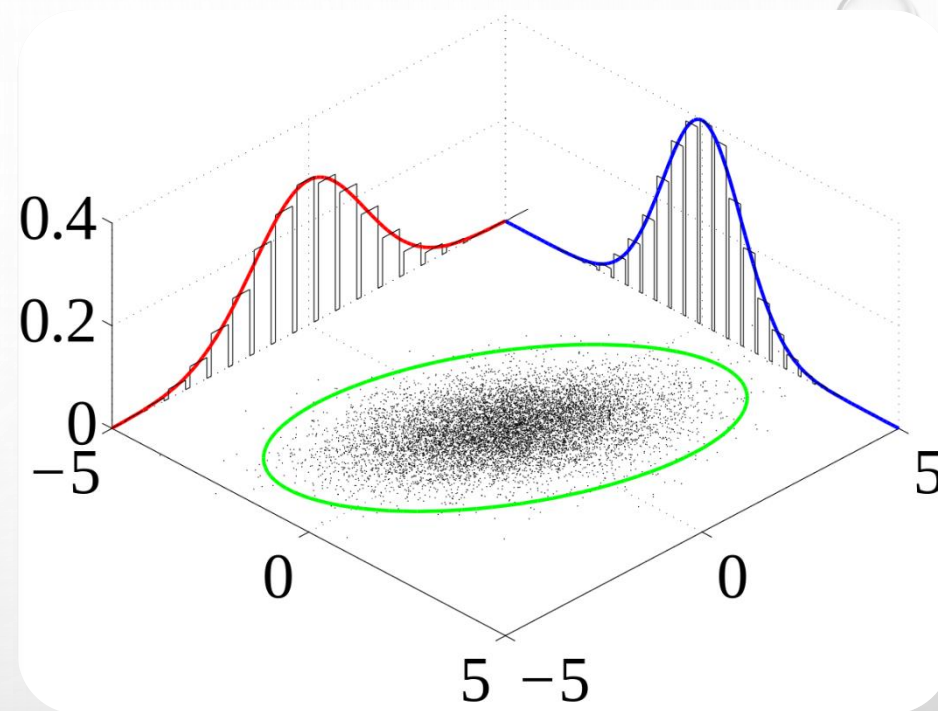
Exercício: qual seria uma boa representação para diferenciar ratos e elefantes?

MODELING

ALGORITMOS BASEADOS EM DENSIDADE

Algoritmos que dependem da **função densidade de probabilidade** dos dados, ou aproximações locais, para determinar a classe de observações fora da amostra de treino.

- 1) Classificador Bayesiano
- 2) Classificador Bayesiano “Naïve”
- 3) K-Vizinhos mais próximos



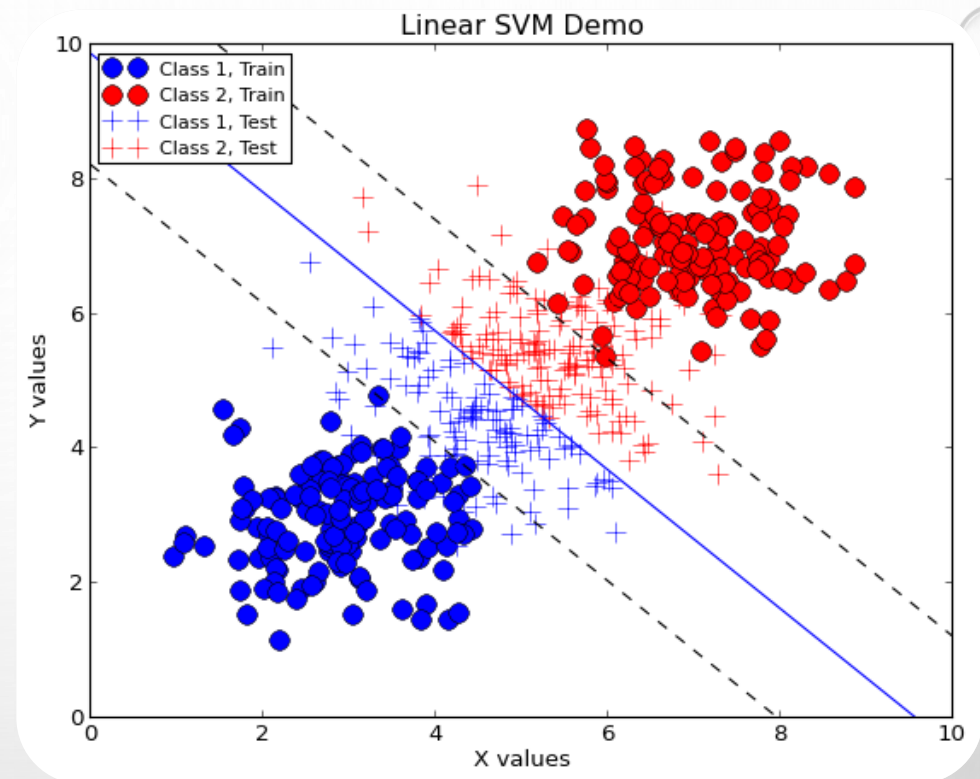
Algoritmos baseados em densidade dependem da **DENSIDADE (!!!)**. Consequentemente, se beneficiam de um **conjunto grande de observações e de baixa esparsidade do espaço de atributos**. O Classificador Bayesiano é considerado o classificador “ótimo”, mas é raramente utilizado, dada a dificuldade de estimar a função densidade de probabilidade dos dados. É normalmente utilizado como benchmark para comparação teórica entre os algoritmos de classificação.

MODELOS FUNCIONAIS

Algoritmos que dependem da **estimação dos parâmetros de uma função** que é utilizada como **superfície de separação** entre as classes.

- 1) Funções Polinomiais
- 2) Regressão Logística
- 3) Máquina de Vetores Suporte
- 4) **Neurônio Sigmoidal / Tangente Hiperbólica**
- 5) Árvores de Decisão

Algoritmos baseados em funções são **mais simples**, usualmente tem um **número menor de parâmetros** e não dependem em armazenar muitos dados para manter uma “memória”, como por exemplo K-vizinhos mais próximos.



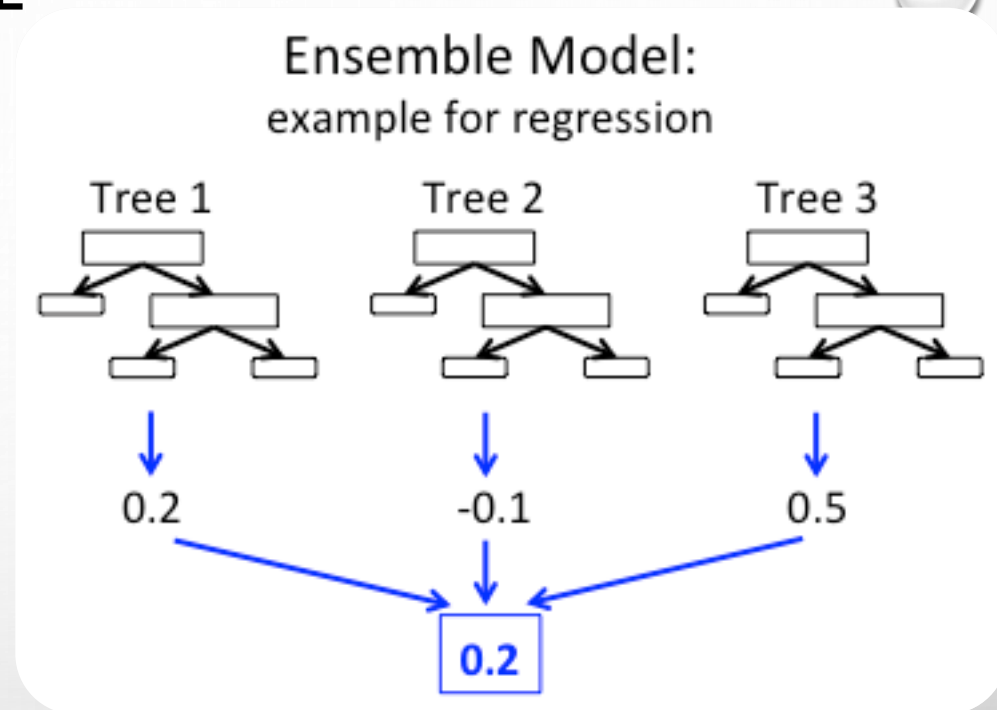
ALGORITMOS BASEADOS EM ENSEMBLE

Algoritmos que **combinam modelos simples**,
usualmente através de **votação ou ponderação**, para
atingir maiores taxas de classificação.

1) Random Forest

2) Boosting

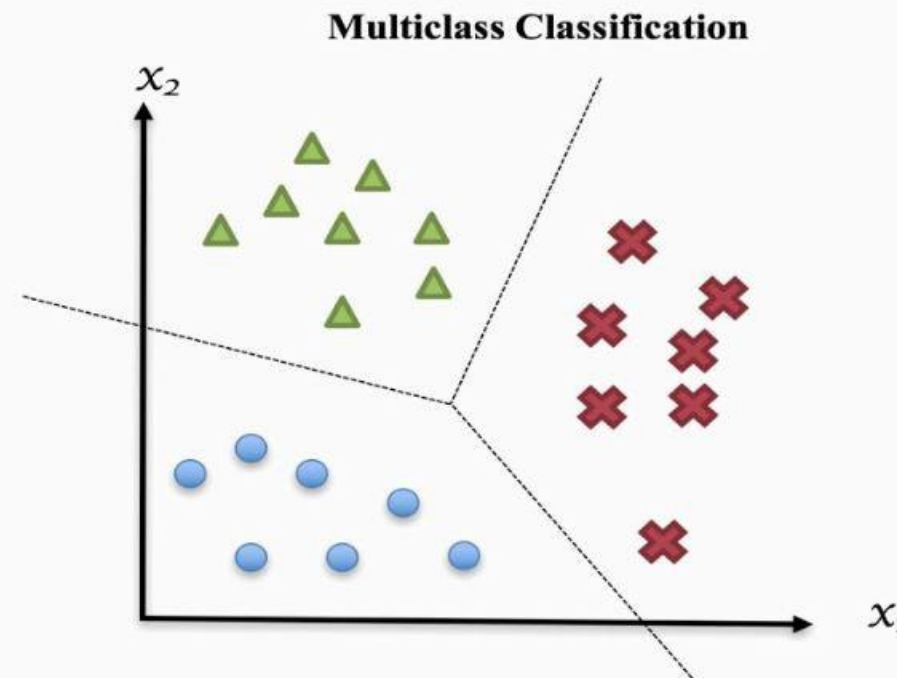
Boa **capacidade de generalização** gerado através de **arranjos complexos** de múltiplos modelos simples de machine learning.



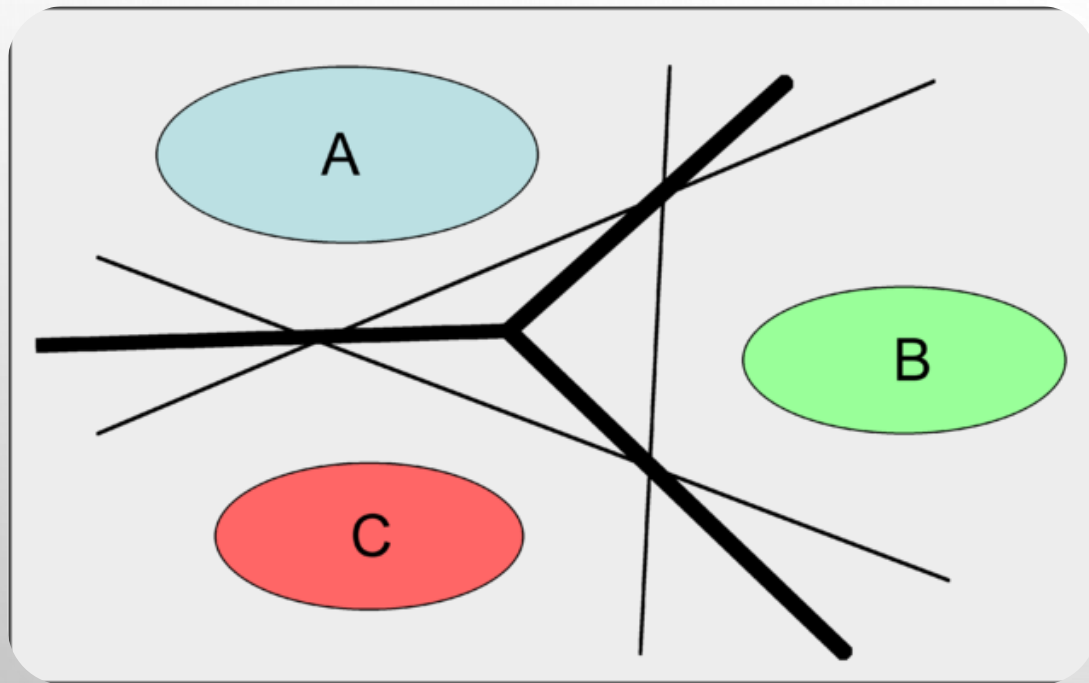
ALGORITMOS BASEADOS EM ENSEMBLE

- **Modelos Multiclasse**

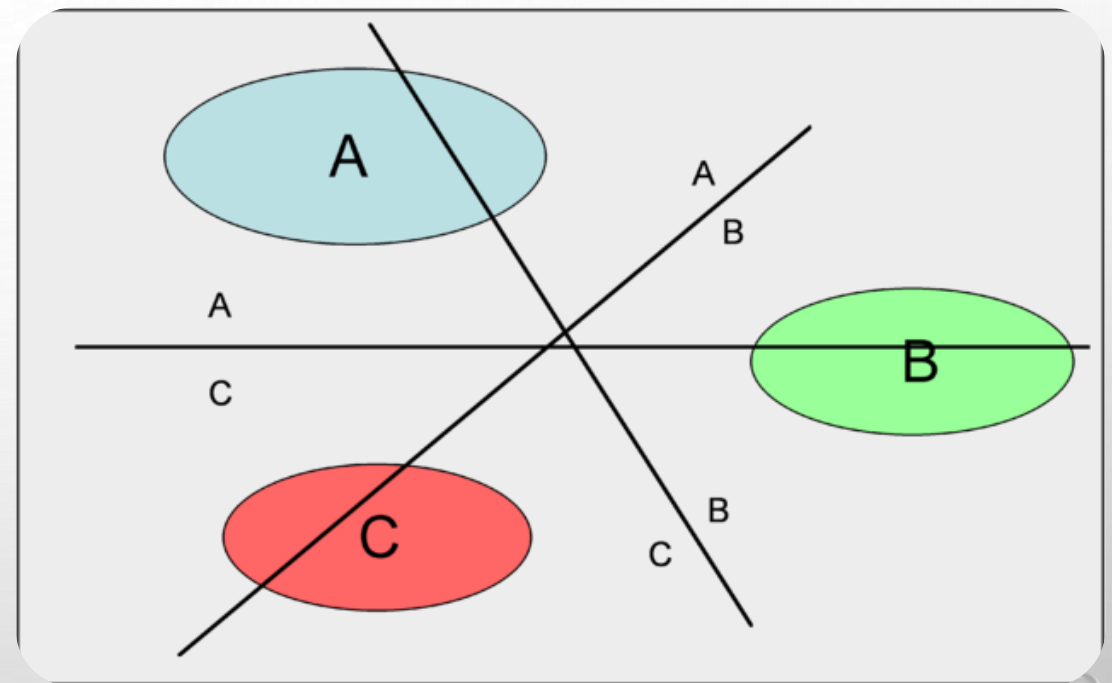
- Discriminar múltiplos objetos em paralelo.
- Ensembles podem ser utilizados para especializar modelos.
- Alguns modelos são naturalmente multiclasse, como redes neurais.



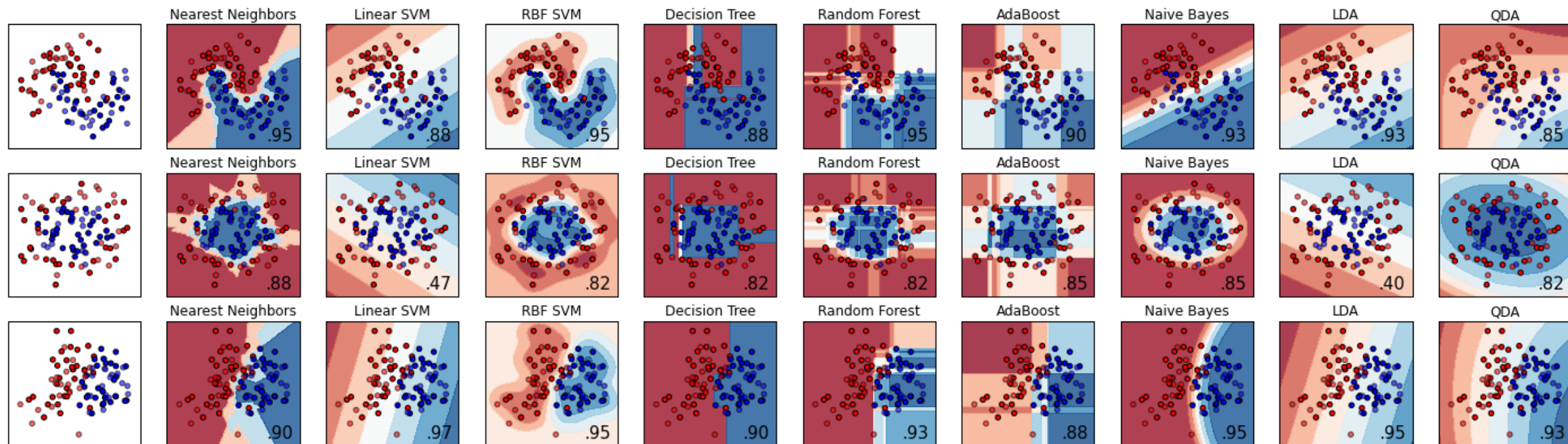
ENSEMBLES BÁSICOS



ONE AGAINST ALL



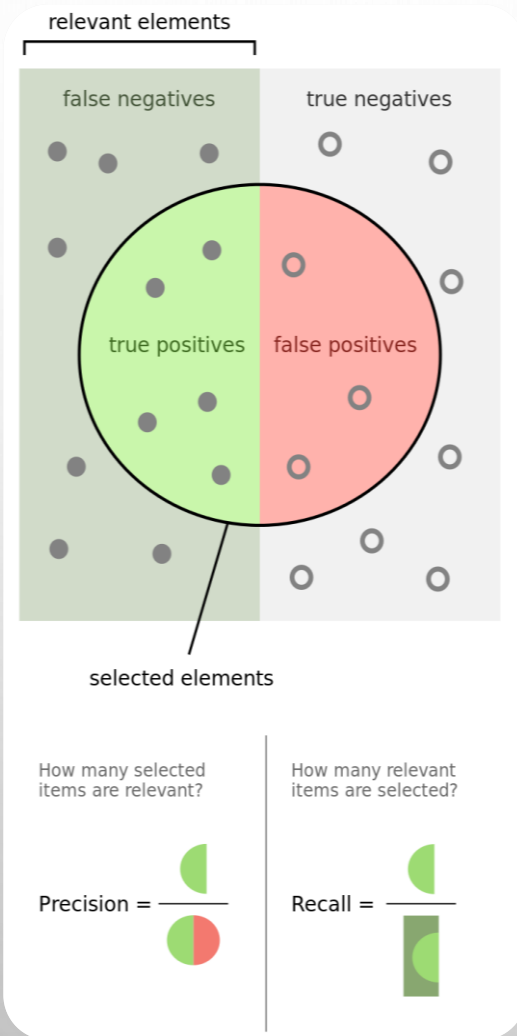
ONE AGAINST ONE





EVALUATION

FIGURAS DE MÉRITO CLASSIFICAÇÃO



Acurácia

- $(TP+TN)/(P+N)$

Taxa de Erro

- $1 - \text{Acurácia}$

Sensibilidade (Recall)

- $TP/(TP+FN)$

Especificidade

- $TN/(TN+FP)$

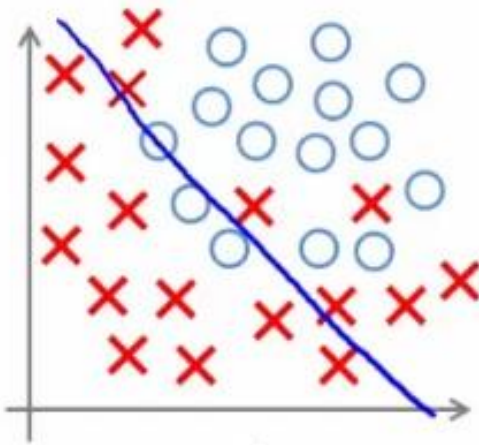
Precisão

- $TP/(TP+FP)$

Produto Sp

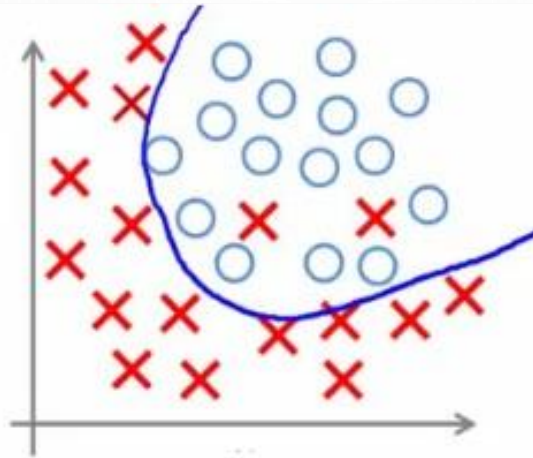
- $\text{SQRT}[\text{SQRT}(R1 * R2) * (R1 + R2)/2]$

CAPACIDADE E GENERALIZAÇÃO

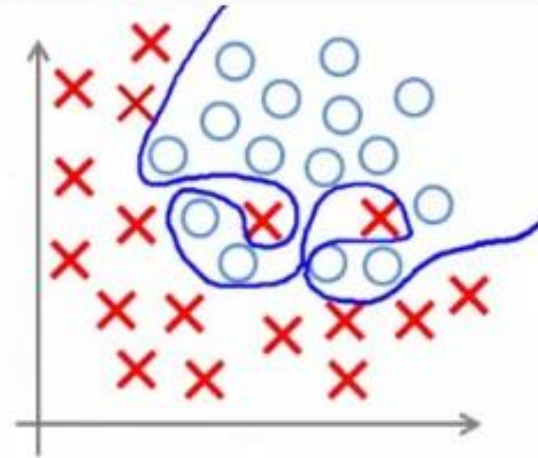


Under-fitting

(too simple to
explain the
variance)



Appropriate-fitting



Over-fitting

(forcefitting -- too
good to be true)

GENERALIZAÇÃO: IDENTIFICANDO OS HIPER- PARÂMETROS ÓTIMOS

LEAVE ONE OUT

- Uma única observação é deixada de fora a cada treinamento. N treinamentos são realizados para calcular a estatística de erro.

K FOLDS

- Amostra é dividida em K conjuntos. K treinamentos são realizados, mantendo um conjunto como fora-da-amostra.

BOOTSTRAPPING

- O algoritmo itera, amostrando aleatoriamente M observações, para a quantidade Q desejada de treinamentos.

TREINAMENTO K FOLDS

- TREINAMENTO UTILIZANDO K PARTIÇÕES, COM DADOS DAS CLASSES BALANCEADOS.
- CADA TREINAMENTO É REALIZADO PARA EXPLORAR UMA CONFIGURAÇÃO DE HIPERPARÂMETROS DO MODELO.
- 4 PARTIÇÕES SÃO USADAS PARA TREINAR O MODELO, 1 PARTIÇÃO É UTILIZADA PARA MENSURAR O DESEMPENHO FORA DA AMOSTRA (GENERALIZAÇÃO).
- UMA ESTATÍSTICA DA FIGURA DE MÉRITO É SELECIONADA PARA MEDIR A QUALIDADE DE CADA CONFIGURAÇÃO DE HIPERPARÂMETRO.

Teste

Treino

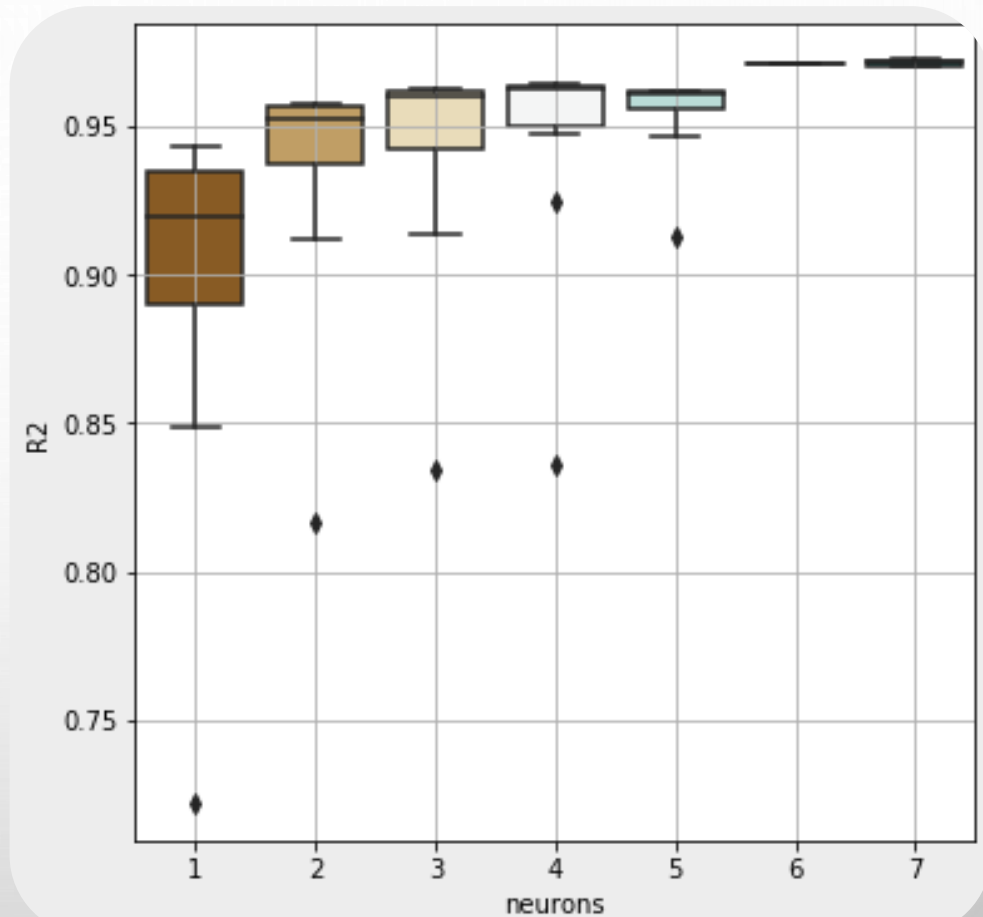
Treino

Treino

Treino

K-FOLDS - EXEMPLO

- **Iteração dos hiperparâmetros**
- **Seleção da Figura de Mérito**
- **Seleção da Estatística de Ganho**



The background is a light gray gradient. In the top-left and bottom-right corners, there are several realistic water droplets of various sizes, some overlapping. A faint, circular watermark is visible in the upper center of the page.

PARTE 2 : PRÁTICA

AMBIENTE PYTHON



4. Variáveis
Aleatórias



1. Editor de Código



5. Visualização



6. Machine
Learning



2. Gestor de Ambiente



3. Ambiente
Python do Projeto



3. Notebook
Dinâmico

PROBLEMA DE NEGÓCIO

Características das flores

Largura & comprimento da pétala

Largura & comprimento da sépala



Iris Setosa



Iris Versicolor



Iris Virginica

Iris Setosa

Iris Versicolor

Iris Virginica

REPRESENTAÇÃO



Iris Setosa



Iris Versicolor

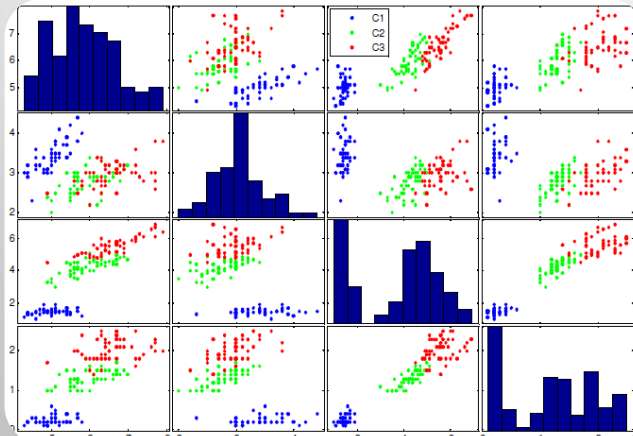


Iris Virginica

Características das flores

Largura & comprimento da pétala

Largura & comprimento da sépala



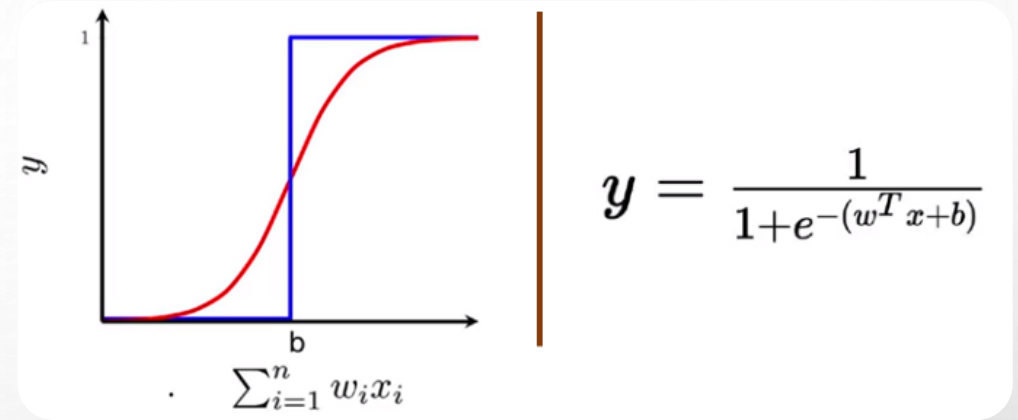
<http://archive.ics.uci.edu/ml/datasets/Iris>

Espaço de
atributos com
4 dimensões!

MODELAGEM

REDE NEURAL FEED FORWARD

- REPRESENTAÇÃO: 2 ATRIBUTOS
- META-PARÂMETROS: 1 NEURÔNIO SIGMOIDE
- TREINAMENTO: BASE DE TREINO COMPLETA.
 - BASE DE VALIDAÇÃO PERCENTUAL
 - CRITÉRIO DE PARADA: NÚMERO DE ÉPOCAS
- ALGORITMO RMSPROP
 - BATELADA COMPLETA
 - TAXA DE APRENDIZADO
- FIGURA DE MÉRITO: PRECISÃO



$$\Delta w_{ij} = (\eta * \frac{\partial E}{\partial w_{ij}})$$

weight increment learning rate weight gradient

$$\Delta w_{ij} = (\eta * \frac{\partial E}{\partial w_{ij}}) + (\gamma * \Delta w_{ij}^{t-1})$$

momentum factor weight increment, previous iteration



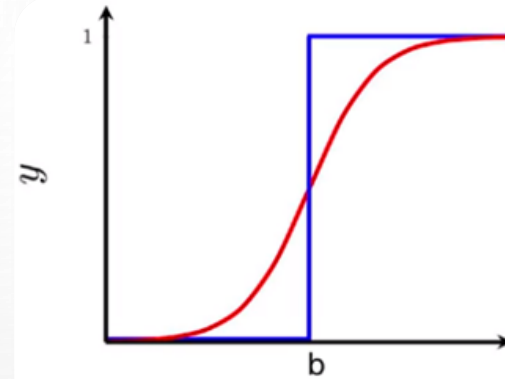
Iris Setosa



Iris Versicolor



Iris Virginica



$$y = \frac{1}{1 + e^{-(w^T x + b)}}$$

$$\cdot \sum_{i=1}^n w_i x_i$$

$$\cdot \sum_{i=1}^I m_i x_i$$

CLASSIFICADOR IRIS

EXERCÍCIO: NEURÔNIO SIGMÓIDE

PRÓXIMA AULA:
CLASSIFICAÇÃO REDE
NEURAL UMA CAMADA
OCULTA